CS 201 Homework 03
CHANGEME: HW 03

Stefano Fochesatto

September 30, 2019

Source Code Link: `https://github.com/StefanoFochesatto/cs201/tree/master/Homeworks/HW3`

This homework took approximately 02 hours to complete.

1 Design

For the boxer function I had a very simple design, I basically had a for loop that for the vertical length of the box. The a nested for loop for each type of output string. There was three types of output strings, one that was all ”*” one that was composed of ”*” and spaces, and another that was composed of ”*”, spaces, and the input string. For the collatz program I only made a function that evaluates the collatz function, but I could have made a separate function which iterates the the collatz function to create the vector for the sequence. For the quadratic function we know that the roots are determined by the discriminant so it is easy to create of else statement to get the right solution type.

2 Post Mortem

I spent a lot of time Implementing the if else statements in the boxer program. The problem was that i was putting a new line inside of every nested for loop so I kept getting three really long strings. The whole section of code where I build the collatz sequence I think can be condensed into one line of code, but it kept giving me an error and I couldn't figure it out.

3 Answers to Questions

In this section, you will write the answers to the questions in the homework assignment.

- Name four major types of errors and briefly define each one?
  Compile-time errors, Link-time errors, Run-time errors, and Logic errors. Compile-time errors are errors found by the compiler, such as syntax errors or type errors. Link-time errors are errors that are found by the linker when trying to combine object files into an executable program. Run-time errors are errors found by checks in a running program. Logic errors are errors found that are created by faulty logic.

- What kinds of errors do we ignore in student programs?
  Errors with misbehaving hardware and misbehaving system software.

- What guarantees should every completed project offer?
  Programs should be properly debugged and tested. Multiple commits and it should always compile.

- List three approaches we can take to eliminate errors in programs and produce acceptable software?
  There are a number of things that we can do to help eliminate errors in our code. First is commenting code, this allows the programmer to think through all the logical errors in the code, and it even helps eliminate syntax errors. Breaking long sections of code into smaller functions allows

us to partition our code, and to make our code modular. Using a consistent layout of code, helps us keep our code organised clear.

- Why do we hate debugging?
  Some people hate debugging because it means we aren't as methodical and logical as we can be. What is important to keep in mind is a less experienced programmer with a healthy attitude toward fixing errors, is far more productive than an experienced programmer with a bad attitude towards fixing errors.

- What are the steps in debugging a program?
  1. Get the program to compile
  2. Get the program to link
  3. Get the program to do what you want.

- Why does commenting help debugging?
  Commenting code allows the programmer to think through all the logical errors in the code, and it even helps eliminate syntax errors

- How does testing differ from debugging?
  The purpose of debugging is to correct those bugs found during testing. While The purpose of testing is to find bugs and errors when the program is already executing.

4 Sample Output

Listing 1: Sample Program Output

```
Please enter a string: Stefano
Please enter the width of your border: 2
```

```
*************
*************
**         **
** Stefano **
**         **
*************
*************

Please enter a string: exit
Program ended with exit code: 0
```

5 boxer

```
#include <iostream>
#include <string>
#include <vector>
#include "boxer.hpp"


int main(int argc, const char * argv[]) {


    while (true){

    std::cout << "Please enter a string: ";

    std::string user_string;

    std::cin >> user_string;

        if (user_string == "exit"){
            break;
        }

    std::cout << "Please enter the width of your border: ";

    int border;

    std::cin >> border;

        if (border<=0){
            std::cout << "Please enter a positive width for your border: \n ";
            std::cin >> border;
        }
    boxer(user_string,border);

    }


    return 0;
}
```

6 collatz

Listing 2: Sample Program Output

```
Please enter an int: 61
61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4
   2 1 Program ended with exit code: 0
```

7 boxer

```
#include <iostream>
#include <vector>
#include "collatz.hpp"


int main(int argc, const char * argv[]) {
    std::vector<int> collatz_vect;


    std::cout << "Please enter an int: ";
    int collatz_start;
    std::cin >> collatz_start;
    collatz_vect.push_back(collatz_start);
    while (true){
        if (collatz_vect.back() == 1){
            break;
        }
        else{
            int collatz_current = collatz_vect.back();
            int collatz_next = collatz(collatz_current);
            collatz_vect.push_back(collatz_next);
        }
    }

    for(int i=0; i < collatz_vect.size(); i++)
       std::cout << collatz_vect.at(i) << ' ';


    return 0;
}
```

8 quadratic

Listing 3: Sample Program Output

```
Please enter the coeffecients to the quadratic
   funciton of the form ax^2+bx+c = 0
a: 2
b: 4
c: 2
Roots are real and the same.
x_[1] = x_[2] =-1
Please enter the coeffecients to the quadratic
   funciton of the form ax^2+bx+c = 0
a:
```

9 boxer

```
int main(int argc, const char * argv[]) {

    while (true){

    double a,b,c;
    std::cout << "Please enter the coeffecients to the quadratic funciton of the for
    std::cout << "a: ";
    std::cin >> a;

    std::cout << "b: ";
    std::cin >> b;

    std::cout << "c: ";
    std::cin >> c;
        if ((a == 0)&&(b == 0)&&(c == 0))
        {
            break;
        }

    quadratic(a,b,c);

    }





    return 0;
}
```